



字典树

河南省实验中学
信息技术组

字典树

概念

初始化

插入

检索

例题

前缀统计

最大异或对

最长异或路径

练习

字典树 Trie

河南省实验中学信息技术组

2026年02月03日



字典树

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

- 字典树 (Trie) 是一种用于实现字符串存储和快速检索的多叉树结构。
- 从根结点到任意结点经过的分支所代表的字符可以构成字符串。

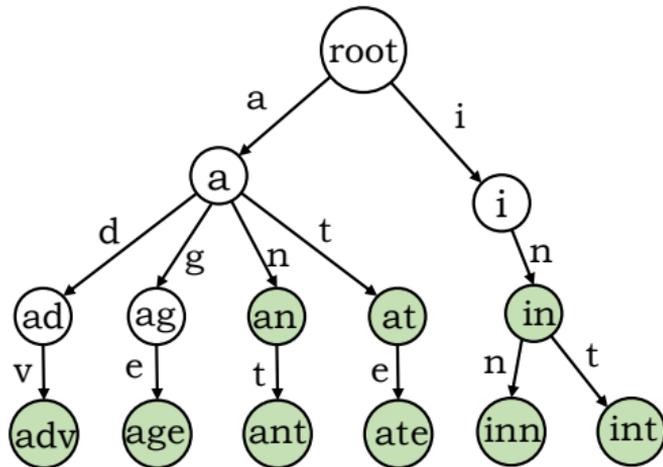


图: 字典树



字典树

字典树

河南省实验中学
信息技术组

字典树

概念

初始化

插入

检索

例题

前缀统计

最大异或对

最长异或路径

练习

- 对于存在公共前缀的字符串集合，可以节省存储空间和查询前缀的时间。
- 一般需要在结点上保存额外信息，例如标记结点是否是字符串或者字符串出现的次数。

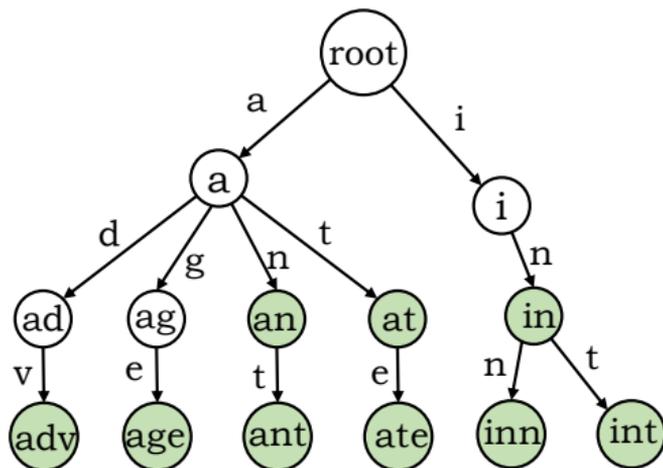


图: 字典树 (结点有颜色标志其为一个字符串)



初始化

字典树

河南省实验中学
信息技术组

字典树

概念

初始化

插入

检索

例题

前缀统计

最大异或对

最长异或路径

练习

- 一个空字典树仅包含一个根结点，该结点的字符指针均指向空 (0)。

```
1 // 假设字符串由小写字母构成 则最多有 26 个孩子
2 int trie[N][26];
3 // 初始时只有一个根结点
4 int tot = 1;
5 // 结点是否是一个字符串的结尾 (是一个单词)
6 bool word[N];
7 // 全部初始化为 0
8 void init()
9 {
10     memset(trie, 0, sizeof(trie));
11     memset(word, false, sizeof(word));
12 }
```

- 空间复杂度: $O(NC)$, 其中 N 是结点数, C 是字符集大小。



插入

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

当需要插入字符串 s 时，令 p 表示根结点，扫描 s 中的每个字符 c ：

- ① 若 p 的 c 字符指针指向一个已经存在的结点 q ，则令 $p = q$ ；
- ② 若 p 的 c 字符指针指向空，则新建一个结点 q ，令 p 的 c 指针指向 q ，然后令 $p = q$ 。

当字符串 s 中的字符扫描完毕时，在当前结点 p 上标记它是一个字符串的结尾。



插入

字典树

河南省实验中学
信息技术组

字典树

概念

初始化

插入

检索

例题

前缀统计

最大异或对

最长异或路径

练习

```
1 // 插入字符串
2 void insert(string s)
3 {
4     int p = 1; // 当前结点为根结点
5     for(int i = 0; i < s.size(); ++i)
6     {
7         int c = s[i] - 'a'; // 根据字符找到对应的字符指针
8         // 若对应的字符指针为空，则新建一个结点
9         if(trie[p][c] == 0) trie[p][c] = ++tot;
10        // 指向下一个结点
11        p = trie[p][c];
12    }
13    // 标记字符串结尾
14    word[p] = true;
15 }
```



检索

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

当需要检索字符串 s 是否在字典树中时，令 p 表示根结点，扫描 s 中的每个字符 c ：

- ① 若 p 的 c 字符指针指向空，则说明 s 不在字典树中，结束；
- ② 若 p 的 c 字符指针指向一个已经存在的结点 q ，则令 $p = q$ 。

当字符串 s 中的字符扫描完毕时，若当前结点被标记为一个字符串的末尾，则说明 s 在字典树中，否则 s 不在字典树中。



检索

字典树

河南省实验中学
信息技术组

字典树

概念

初始化

插入

检索

例题

前缀统计

最大异或对

最长异或路径

练习

```
1 // 检索字符串是否在字典树中
2 bool search(string s)
3 {
4     int p = 1; // 当前结点为根结点
5     for(int i = 0; i < s.size(); ++i)
6     {
7         int c = s[i] - 'a'; // 根据字符找到对应的字符指针
8         // 若对应的字符指针为空，则说明不存在
9         if(trie[p][c] == 0) return false;
10        // 指向下一个结点
11        p = trie[p][c];
12    }
13    // 最后一个结点是否被标记为字符串结尾
14    return word[p];
15 }
```



【例】前缀统计

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

【题目描述】

给定 n 个字符串 s_1, s_2, \dots, s_n ，接下来进行 m 次询问，每次询问给定一个字符串 t ，求 $s_1 \sim s_n$ 中有多少个字符串是 t 的前缀。输入字符串的总长度不超过 10^6 ，仅包含小写字母。

【输入格式】

第一行输入两个正整数 n, m 。

接下来 n 行，每行一个字符串 s_i 。

接下来 m 行，每行一个字符串 t 表示询问。

【输出格式】

对于每个询问，输出一行一个整数表示询问。

【样例输入】

```
3 2
ab
bc
abc
abc
efg
```

【样例输出】

```
2
0
```



前缀统计

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

- 把这 n 个字符串插入一棵字典树，字典树的每个结点上存储一个整数 cnt ，记录该结点是多少个字符串的末尾结点。
- 对于每个询问，在字典树中检索 T ，在检索过程中累加途径中每个结点的 cnt 值，就是该询问的答案。

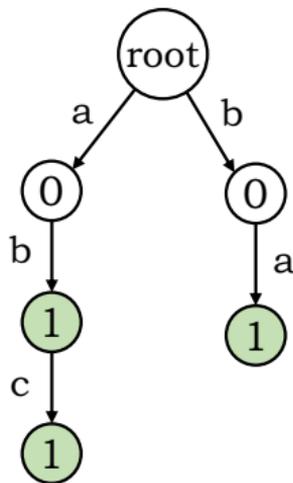


图: 样例



【例】前缀统计

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

```
1 const int N = 1e6 + 10;  
2 int trie[N][26], tot = 1;  
3 int cnt[N];  
4  
5 int n, m; cin >> n >> m;  
6 for(int i = 1; i <= n; ++i)  
7 {  
8     string s; cin >> s;  
9     insert(s);  
10 }  
11 for(int i = 1; i <= m; ++i)  
12 {  
13     string t; cin >> t;  
14     cout << search(t) << "\n";  
15 }
```



【例】前缀统计

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

```
1 void insert(string s)
2 {
3     int p = 1; // 当前结点为根结点
4     for(int i = 0; i < s.size(); ++i)
5     {
6         int c = s[i] - 'a'; // 根据字符找到对应的字符指针
7         // 若对应的字符指针为空, 则新建一个结点
8         if(trie[p][c] == 0) trie[p][c] = ++tot;
9         p = trie[p][c]; // 指向下一个结点
10    }
11    ++cnt[p]; // 单词数加 1
12 }
13 int search(string s)
14 {
15     int p = 1, ans = 0; // 当前结点为根结点
16     for(int i = 0; i < s.size(); ++i)
17     {
18         int c = s[i] - 'a'; // 根据字符找到对应的字符指针
19         if(trie[p][c] == 0) break; // 若对应的字符指针为空, 后面不会有前缀
20         p = trie[p][c]; // 指向下一个结点
21         ans += cnt[p]; // 是单词 答案加上计数
22     }
23     return ans;
24 }
```



【例】最大异或对

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

【题目描述】

给定 n 个整数 a_1, a_2, \dots, a_n 中选两个数进行异或运算，得到的结果最大是多少？

【输入格式】

第一行一个正整数 n ($n \leq 10^5$)。

第二行 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i < 2^{32}$)。

【输出格式】

一行一个整数表示答案。

【样例输入】

```
3
1 2 3
```

【样例输出】

```
3
```

【样例解释】

输入的三个数分别为 1、2、3，则最大的异或对为 $1 \text{ xor } 2 = 3$ 。



【例】最大异或对

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

- 算法 1: 枚举
- 枚举任意两个数 $a_i (1 \leq i \leq n)$ 和 $a_j (i < j \leq n)$, 求出 $a_i \text{ xor } a_j$, 比较得出最大值。
- 时间复杂度: $O(N^2)$ 。



【例】最大异或对

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

- 算法 2: 字典树
- 把每个整数看做长度为 32 的二进制 01 串, 把 $a_1 \sim a_n$ 对应的 32 位二进制串插入一棵字典树 (其中最低二进制位为叶子结点)。
- 对于每个 a_i 对应的 32 位二进制串, 在字典树中进行一次检索, 只是每一步都**尽力**沿着“与 a_i 的当前位相反的字符指针”向下访问。
- 若“与 a_i 的当前位相反的字符指针”为空, 则只好沿着“与 a_i 的当前位相同的字符指针”向下访问。
- 访问到叶子结点时生成的二进制序列对应的值即为与 a_i 做异或运算结果最大的值。
- 时间复杂度: $O(32N)$ 。



【例】最大异或对

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

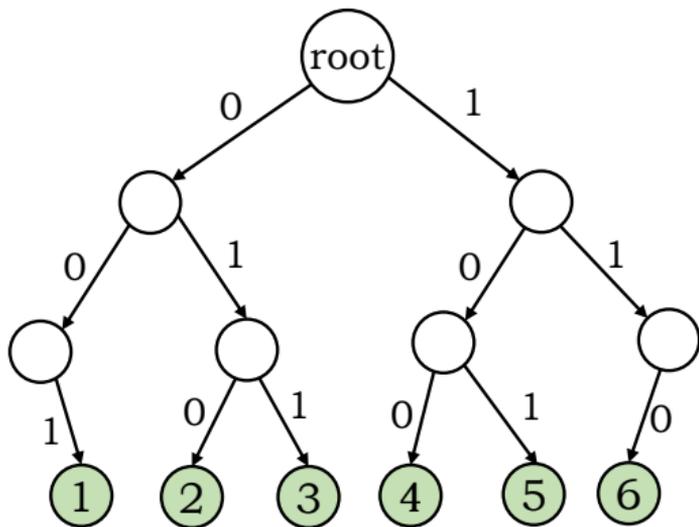


图: 样例 (用三位二进制表示)



【例】最大异或对

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

```
1 const int N = 1e5 + 10;  
2 int trie[N << 5][2], tot = 1;  
3 int a[N];  
4  
5 for(int i = 1; i <= n; ++i)  
6     insert(a[i]);  
7 int ans = -1;  
8 for(int i = 1; i <= n; ++i)  
9     ans = max(ans, search(a[i]));  
10 cout << ans;
```



【例】最大异或对

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

```
1 void insert(int num)
2 {
3     int p = 1;
4     for(int i = 30; i >= 0; --i)
5     {
6         int t = (num >> i) & 1; // 获取 num 的第 i 位
7         if(trie[p][t] == 0) trie[p][t] = ++tot; // 新建结点
8         p = trie[p][t];
9     }
10 }
11 int search(int num)
12 {
13     int p = 1, ans = 0;
14     for(int i = 30; i >= 0; --i)
15     {
16         int t = (num >> i) & 1; // 获取 num 的第 i 位
17         if(trie[p][t ^ 1]) {
18             ans |= (1 << i); // 答案的第 i 位置为异或结果 1
19             p = trie[p][t ^ 1]; // 向当前位相反的路径走
20         }
21         else p = trie[p][t]; // 与当前位相反的路径不存在 只能沿着当前路径走
22     }
23     return ans;
24 }
```



【例】最长异或路径

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

【题目描述】

输入的三个数分别为 1、2、3，则最大的异或对为 $1 \text{ xor } 2 = 3$ 。给定一棵 n 个结点的树，树上的每条边都有一个权值。从树中选择两个顶点 x 和 y ，把从 x 到 y 的路径上的所有边权异或起来，得到的结果最大是多少？

【输入格式】

第一行一个正整数 n ($n \leq 10^5$)。

接下来 $n - 1$ 行，每行三个整数 x, y, w ($1 \leq x, y \leq n, 0 \leq w < 2^{31}$)，表示顶点 x 和顶点 y 之间有一条边权重为 w 。

【输出格式】

输出一个整数，表示最大的异或长度。



【例】最长异或路径

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

【样例输入】

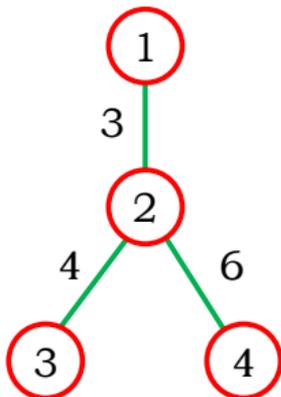
```
4
1 2 3
2 3 4
2 4 6
```

【样例输出】

```
7
```

【样例解释】

最长的异或路径为 $1 \rightarrow 2 \rightarrow 3$ ，结果为 $3 \text{ xor } 4 = 7$ 。





【例】最长异或路径

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

- 算法 1: 枚举
- 对于每个顶点, 以其为根, DFS 或 BFS 求出其到所有孩子结点的路径异或值, 然后求出最大值。
- 设 $d[x]$ 表示根结点到 x 的路径上所有边权的异或值, 显然有:

$$d[x] = d[p(x)] \text{ xor } w(x, p(x))$$

$p(x)$ 为 x 的双亲结点。

- 对于每个顶点的最大值, 再取最大值即可得出答案。
- 时间复杂度: $O(N^2)$ 。



【例】最长异或路径

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

- 算法 2:
- 任选一个顶点为根，DFS 或 BFS 求出其到所有孩子结点的路径异或值。
- 设 $d[x]$ 表示根结点到 x 的路径上所有边权的异或值，显然有：

$$d[x] = d[p(x)] \text{ xor } w(x, p(x))$$

$p(x)$ 为 x 的双亲结点。

- 可以发现，树上 x 到 y 的路径上所有边权的异或值结果等于 $d[x] \text{ xor } d[y]$ 。
- 问题转化为从 $d[1] \sim d[n]$ 这 n 个数中选出两个，异或值最大，问题转化为最大异或对问题。
- 时间复杂度： $O(N)$ 。



【例】最长异或路径

字典树

河南省实验中学
信息技术组

字典树

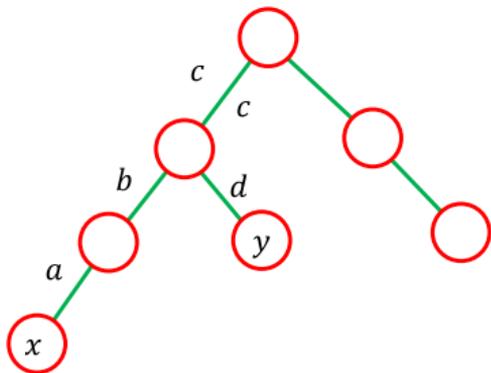
概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

- 树上 x 到 y 的路径上所有边权的异或值结果等于 $d[x] \text{ xor } d[y]$ 。
 - 顶点 x 到顶点 y 的路径经过根结点，此时明显答案为 $d[x] \text{ xor } d[y]$ 。
 - 顶点 x 到顶点 y 的路径不经过根结点，那么它们一定在根结点的子树上， x 到根的路径与 y 到根的路径会有重叠的部分，而这部分恰被计算两次，由异或的性质 ($a \text{ xor } a = 0$)，不会产生影响，答案仍为 $d[x] \text{ xor } d[y]$ 。
 - 如下图， $d[x] \text{ xor } d[y] = (a \text{ xor } b \text{ xor } c) \text{ xor } (c \text{ xor } d) = a \text{ xor } b \text{ xor } (c \text{ xor } c) \text{ xor } d = a \text{ xor } b \text{ xor } 0 \text{ xor } d = a \text{ xor } b \text{ xor } d$ 。





练习

字典树

河南省实验中学
信息技术组

字典树

概念
初始化
插入
检索

例题

前缀统计
最大异或对
最长异或路径

练习

- 单词查找树 [NOI 2000](COGS 293)
- 前缀统计 (COGS 3423)
- 通讯录 (COGS 3440)
- 昵称 (COGS 788)
- 有道搜索框 (COGS 647)
- strcmp() 函数 (COGS 2695)
- 单词默写 (COGS 812)
- Asm.Def 谈笑风生 (COGS 2082)
- 最大异或对 (COGS 3424)
- 最长异或路径 (COGS 3474)